International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN (E): 2249-7943
Vol. 4, Issue 4, Aug 2014, 77-84

OPTIMIZED FAULT TOLERANCE IN REAL TIME CLOUD COMPUTING

JATINDER KAUR & MEENAKSHI SHARMA

Department of Computer Science Engineering, Sri Sai Institute of Engineering and Technology,

Badhani, Pathankot, Punjab, India

ABSTRACT

Fault tolerance is one of the processes which are very essential due to risk attached to it. In case of any fault occurrence in real environment, loss occurred in term of finance and reputation and even in term of user experiences. So there is an increased need to tolerate the fault for such type of systems to be used with cloud computing structure. For the development of this process and to provide better fault tolerance mechanism while using cloud computing infrastructure for real time applications, we will present an optimized fault tolerance of real time applications running at cloud infrastructure. Reliability and load of the tasks will be the criteria for deciding the degree of fault tolerance for cloud system.

KEYWORDS: Fault Tolerance, Reliability, Cloud Computing and Real Time Computing

INTRODUCTION

Real time computing for management of resources is a challenging process to follow for the systems which are centralized in nature. Cloud computing is commonly applicable to this category due to limitations and policies applicability for computing process. From last decade cloud services is provide high end computing for different applications for different domains. Industrial processes are opting cloud computing process but many of the normal units and people are not using this computing because, they are not aware of the advantages of using these systems. Real time applications are very important for users but users are reluctant for adoption of real time applications. Cloud services are very useful for real times based applications for different domain. Application variation from basic smartphones to small set industrial applications is very essential. Normally, real-time system is any information processing unit which provide response to input provide by external source, with limitation of time. Due to this dependency and realness of the applications, correct results are needed rather than just logical results. Any type of failure is not acceptable as failure of single unit would rise to other unit failure too. Timeline and fault tolerance are two parameters which decide the performance of real time applications. Time limitation is always an essential part of the real time processes and need to be executed within decided deadline with strict time constraints. Fault tolerance is as important as time in real time applications and degree of tolerance is always decide the effectiveness of real time applications. Error occurrence increases if we integrate cloud computing to real time applications due to virtualization concepts in cloud services which are more tends towards logical performance and job processing and sensing dependent.

Fault tolerance is one of the processes which are very essential due to risk attached to it. In case of any fault occurrence in real environment, loss occurred in term of finance and reputation and even in term of user experiences. So there is an increased need to tolerate the fault for such type of systems to be used with cloud computing structure.

For the development of this process and to provide better fault tolerance mechanism while using cloud computing infrastructure for real time applications, we will present an optimized fault tolerance of real time applications running at cloud infrastructure. The previous study of the similar mechanism is also based on fault tolerance and it is based on replication with reliability threshold count. Previous work performed replication in term of software variation running on multiple virtual machines. The reliability of the virtual machines is adaptive, which changes after every computing cycle [1]. If a virtual machine manages to produce a correct result within the time limit, its reliability increases [1]. And if it fails to produce the result within time or correct result, its reliability decreases. The proposed infrastructure in previous related study is displayed below in figure 1.

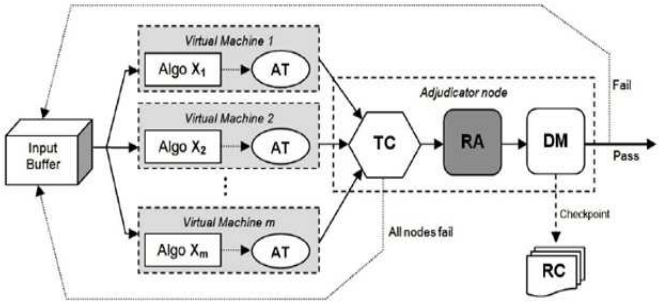

Figure 1: Adaptive Fault Tolerance Mechanism with Reliability Process [1]

Adaptive fault tolerance in previous study [1] provided mechanism of checking the reliability of virtual machines and decision node is responsible for deciding the reliability of the virtual machine which will lead to the retaining or replacing virtual machines.

CLOUD COMPUTING FAULT TOLERANCE

In cloud, the latency of virtual machines is unknown. Even if one determines the latency, it can change over the period of time [6]. Node is a virtual and computation can be migrated from one virtual machine Instance to another [19]. The user of real time cloud applications has lose control over the nodes and does not Know where his application is going to be processed. But on the brighter side, cloud has a facility to scale up dynamically. So the faulty node can be removed and new node can be added on demand. These characteristics are different from the existing traditional distributed real time systems [20]. A model for virtual infrastructure Performance and fault tolerance is presented in [21]. A new fault tolerant scheduling algorithm is proposed in [22]. This algorithm incorporates the reliability analysis into the active replication schema, and exploits a Dynamic number of replicas for different tasks. Some pragmatic requirements for highly reliable systems, Highlighted significance and various issues of reliability in different computing environment such as Cloud Computing, Grid Computing, and Service Oriented Architecture are describe in [7]. The Adaptive Fault Tolerance in Real-time Cloud computing (AFTRC) model based upon adaptive reliability assessment of virtual machines in cloud environment and fault tolerance of real time applications running on those VMs was proposed in [6]. AFTRC model tolerates the faults on the basis of reliability of each virtual machine. A virtual machine is selected for computation on the basis of its reliability and can be removed if does not perform well for applications. In AFTRC model, each VM takes the input, executes the application algorithm and produces result. Reliability assessor (RA) module assesses the reliability for each virtual machine is the main core module in AFTRC model. In the beginning the reliability of each virtual machine is 100%. If a processing node manages to produce a correct result within the time limit, its reliability increases. And if the

processing node fails to produce the correct result or result within time, its reliability decreases. The reliability assessment algorithm is more convergent towards failure conditions. It means that decreasing in reliability is more than increasing [6]. In this paper, the proposed algorithm depends on the basic idea (adaptive reliability assessment of virtual machines) of AFTRC model in online cloud task scheduling using an algorithm.

RELATED WORK

The basic idea of ant algorithms is to simulate the foraging behaviour of ant colonies. When an ants group tries to search for the food, they use a special kind of chemical to communicate with each other. That chemical is referred to as pheromone. Initially ants start search their foods randomly [8]. Once the ants find a path to food source, they leave pheromone on the path. An ant can follow the trails of the other ants to the food source by sensing pheromone on the ground. As this process continues, most of the ants attract to choose the shortest path as there have been a huge amount of pheromones accumulated on this path. The total number of ants (m), assumed constant over time, is an important parameter: too many ants would quickly reinforce suboptimal trails and lead to early convergence to bad solutions, whereas too few ants would not produce the expected effects of cooperation because of the process of pheromone decay [10]. The advantages of the algorithm are the use of the positive feedback mechanism, inner parallelism and extensible. The disadvantages are overhead and the stagnation phenomenon, or searching for to a certain extent which mean that all individuals found the same solution exactly, cannot further search for the solution space and make the algorithm converge to local optimal solution [8]. There are many different kinds of ACO algorithm, i.e., Ant Colony System (ACS), Max-Min Ant System (MMAS), Rank-based Ant System (RAS), Fast Ant System (FANT) and Elitist Ant System (EAS) [23]. ACO uses the pseudo-random-proportional rule to replace state transition rule for decreasing computation time of selecting paths and update the pheromone on the optimal path only. It is proved that it helps ants search the optimal path scheduling in cloud environment based ACO algorithms are proposed in [24, 25].

In these methods, the requests are collected; the scheduler considers the approximate execution time for each task and use heuristic approach to possibly make better decision. Tasks are scheduled only at some Cloud Computing Online Scheduling International organization of Scientific Predefined Time this enables batch heuristics to know about the actual execution times of a larger Number of tasks Modified Ant Colony Optimization for Load Balancing (MACOLB) for cloud task scheduling is proposed in [22]. MACOLB algorithm used to find the optimal resource allocation for tasks in the dynamic cloud system, minimize the make span of tasks on the system and increase the performance by balancing the load of the system. An optimized algorithm for virtual machine placement in cloud computing scheduling based on multi-objective ant colony system algorithm in cloud computing is proposed in [2]. Moreover, in [9] an ACO scheduler was introduced to address job scheduling within a Cloud. It is used for online mode. The proposed method is aimed to maximize scheduling throughput to handle all the diversified job requests according to different resources available in a cloud, and minimize the make span of jobs.

SOME ALGORITHMS FOR FAULT TOLERANCE

Algorithm Based on Energy Efficient Optimization Methods

This algorithm is being implemented in Hadoop distributed file system with Energy Management and Regulation also called as Green HDFS. This algorithm concentrates on usage of the resources that are not fully utilized while execution of the environment. Due to fast advancement in technology the old methods of saving energy has been challenging.

The works introduced till now are taken into account with hardware but not with software [5].

Dynamic Priority Scheduling Algorithm (Service Request Scheduling)

This algorithm is applied on three tier containing service providers, resource providers and consumers. This algorithm gives more optimal then First Come First Serve (FCFS) and Static Priority Scheduling Algorithm (SPSA). The consumer response time for services has been tried to reduce in this algorithm as running instance is charged as it runs per unit time. The delays in provider side happens but are not counted under the cost charged to the customer so they need to be reduced. In three tiers there needs to be two scheduling: service request scheduling and resource scheduling [6].

Non-Dominated Sorting Genetic Algorithm II

This algorithm is proposed as a solution for Multi-objective optimization for virtual resources. When one request is made for any resource then the virtual resources scheduling is mapped onto physical resources with proper load balancing which is very complex to achieve. This algorithm is in comparison with rank, random and static algorithm. The layer of virtualization occurs between users and physical layer and it has three characteristics usability, safety an moving. They come from independency of virtualization. The virtual resources are abstracted by making number of instances of actual physical resource nodes with attributes [7].

Optimizing Virtual Machine for High Performance Computing

It is a HPC aware novel scheduler implemented on Open Stack Scheduler. It is topology awaked and homogenously allocating virtual machines. Cloud computing is of the lot of help to those who cannot afford large clusters has replaced supercomputers in some cases. Commodity interconnects performance variability and performance virtualization which indicates that cloud is suited for some HPCs. There are only few efforts on virtual machine algorithms that take into account the HPC. Open stack and Eucalyptus provide a minor effect of HPC. HPC aware strategies (topology awareness and hardware awareness) have been implemented which improves performance by allowing cloud pr oviders to better utilize the infrastructure making more profits. Open stack is a scheduler which selects a physical resource where VM is provisioned [8].

Scheduling with Parallel Genetic Algorithm (PGA)

This algorithm was devised to solve the problem of Unbalance Assignment problem to achieve the maximum efficiency. The existing strategies are not good to handle the scheduling so the GA turns out to be a good choice in case of scheduling. PGA improves performance and scalability. It can be implemented on parallel mainframes and heterogeneous computers. This algorithm helps in finding the best possible scheduling sequence on IaaS (Infrastructure as a Service) cloud giving better results than Rank algorithm, Round Robin algorithm, greedy technique, PBS and SGE [9].

Balance Reduce Algorithm (BAR) (Fault Tolerant)

This algorithm is based on data locality driven reducing network access thus reducing bandwidth usage and job completion time. This algorithm also handles the machine failure. Initial local task allocation in balanced phase takes place and then job execution time can be reduced by matching initial task allocation in reduced phase.

The machine failure is handled by algorithm similar to primary backup approach [10].

Heavy Traffic Optimal Algorithm

The join-the-shortest-queue routing and power-of-two-choices routing with Max Weight scheduling is optimal in throughput and they are queue length optimal in high traffic loads. Calculating the exact queue length is quite difficult so the system in heavy traffic regime (exogenous arrival rate is almost same as boundary of capacity region) was studied. Use of state space collapse (multi dimensional state reduces to single dimension) was there. The algorithm is applied on multiple models supported by multiple servers. This is the stochastic model for load balancing and scheduling in clusters.

The JSQ and Max Weight is throughput optimal and traffic optimal when all servers identical. And also the power of- two-choices is also heavy traffic optimal [11].

Table 1: Comparison of Pass & Fail

Cycle	VM-1		VM-2	
	Status	Reliability	Status	Reliability
Start	-	1.0000	-	1.0000
1	Pass	1.0300	Fail	0.9700
2	Pass	1.0609	Fail	0.9118
3	Pass	1.0927	Fail	0.8297
4	Pass	1.1255	Fail	0.7302
5	Pass	1.1593	Fail	0.6206
6	Pass	1.1941	Fail	0.5089
7	Pass	1.2299	Fail	0.4021
8	Pass	1.2668	Fail	0.3056
9	Pass	1.3048	Fail	0.2231
10	Pass	1.3439	Fail	0.1561

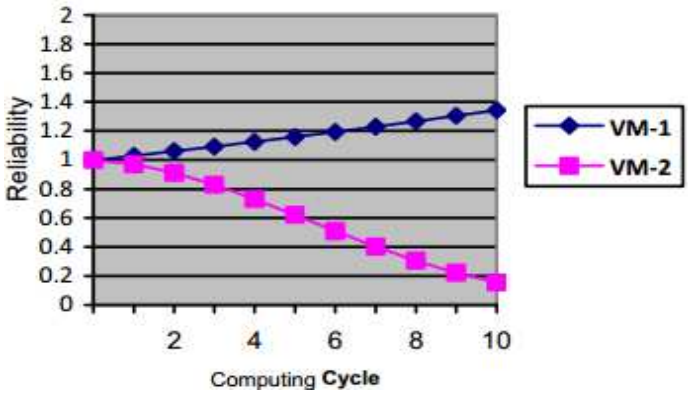

Figure 2: Comparison of Two Virtual Machine Nodes

In the table 2 and figure 3 a scenario for a single node is provided. In this a node continues to be successful for first 5 cycles and then fail for 5 times. Here we can see that convergence towards decrement in reliability is much higher.

Table 2: Pass to Fail Shifting

Cycle	VM Status	Reliability
Start	**-**	**1.0000**
1	Pass	1.0300
2	Pass	1.0609
3	Pass	1.0927
4	Pass	1.1255
5	Pass	1.1593

Table 2: Contd.,		
6	Fail	1.1245
7	Fail	1.0570
8	Fail	0.9619
9	Fail	0.8465
10	Fail	0.7195

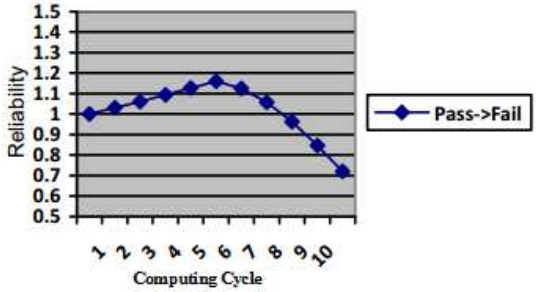

Figure 3: Change in Reliability for a Single Node

PROBLEM DEFINITION

In our research, the system provides fault tolerance mechanism for Virtual machine based on the reliability and the load of the task of the processing nodes. Concept is based on the calculation of results based on the time and number of task executed in particular time then is consider to be fit for process and will provide fault tolerance. If it is not satisfying the proposed criteria then the process is not able to provide fault tolerance in virtual environment. The local reference process is measuring the reliability and load of the virtual machines processes for checking the fulfillment of defined criteria. The experimentation will be done on different virtual machines and can be consider on the real cloud services Like Amazon or Google App or it could also be consider on VMware software

OBJECTIVES

There are many objectives which could be obtained from proposed work. The ideas are simple and effective. To achieve or set proposed scheme and ideas, we will target our objective which is given below “The primary concern and target objective is to find the optimized fault tolerance mechanism based on reliability and task execution load”.

METHODOLOGY

Our research will start with study of parallel computing management in virtual cloud environment based on cloud computing for virtualization in following steps.

1st Phase: This will be the initial stage for the whole process and will contain the basic functionality and collection of information (virtual simulation, basic virtualization functions etc). Layout for comparison will be done in this phase.

2nd Phase: In this stage we will implement the basic scenario for parallel processing structure based on virtualized servers and will provide infrastructure of cloud.

3rd Phase: In this stage we will provide the algorithm of reliability checking based on the number of task executed and execution task load on virtual machines resources.

4th Phase: Decision making will be carried on this step by finding the threshold of the reliability and load in combined way, decision will be taken for whether to assign more virtual machine or to shuffle virtual machines.

5th Phase: Final stage will be for the comparison of the proposed work with already existing work.

REFERENCES

1. Sheheryar Malik, Fabrice Huet, ”Adaptive Fault Tolerance in Real Time Cloud Computing’’, IEEE Conference of World Congress on Services, July 2011, pp.23-34.
2. Anju Bala, Inderveer Chana, “Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing”, International Journal of Computer Science Issues, Vol. 9, Issue.1, January 2012, pp.161-163.
3. Shuai Zhang, “Cloud Computing Research and Development Trend”, Second IEEE International Conference on Future Networks, August 2010, pp.11-13.
4. Ravi Jhawar, “Fault Tolerance Management in Cloud Computing: A System-Level Perspective”, IEEE Systems Journal, Vol. 7, No.2, June 2013, pp.236-242.
5. K. Ganga, “A Fault Tolerent Approach in Scientific Workflow Systems based on Cloud Computing”, Proceedings of the IEEE International Conference on Pattern Recognition, Informatics and Mobile Engineering, February 21-22, pp.117-122.
6. Florin Pop, “Energy-Efficient and Fault Tolerant Methods for Message Delivery in Internet of Things”, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No.5, MAY 2013, pp.563-578.
7. Matthieu Simonin, “An Autonomic and Scalable Management System for Private Clouds”, Thirteenth IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, April 2013, pp. 200-213.
8. Anca Iordache,” Resilin: Elastic Map Reduce over Multiple Clouds”, Thirteenth IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing., April 2013, pp.176-179.